

AMENDMENTS TO THE CLAIMS

1. (currently amended) A system for generating a graphical user interface for an application program, comprising:

one or more business objects that define functions of the application program;

one or more metadata elements defining parameters for the functions of the business object;

a controller configured for invocation by a browser and communicatively coupled to one or more actions, widgets, and panels;

a service object manager coupled to the controller and to the business objects, and configured to supply service object parameter values from the business objects and metadata elements to the actions;

wherein the controller comprises logic configured to receive a user request from a user through the browser used to interact with the application program and to dispatch the user request to one or more of the actions, the controller determining which of the one or more actions is responsible for acting on the user request;

wherein the actions comprise logic configured to interact with the business objects through the service object manager to obtain service object parameter values for the actions;

wherein the controller comprises logic configured to associate the service object parameter values with one of the widgets, to [[place]] arrange the one of the widgets [[in]] into a specified layout within one of the panels, and to generate at runtime an HTML user interface page that includes the panel;

wherein at least one of the widgets has the capability of representing properties of the business objects as HTML.

2. (currently amended) A method of automatically generating a consistent user interface for an application program, the method comprising the computer-implemented steps of:
- receiving one or more business objects that each define a user action for the application program;
 - receiving one or more metadata elements defining parameters for the user actions of the business object;
 - invoking a controller that is communicatively coupled to one or more actions, widgets, and panels;
 - receiving a user request from a user through a browser used to interact with the application program and dispatching the user request to one or more of the actions;
 - the controller determining which of the one or more actions is responsible for acting on the user request;
 - obtaining, using the actions, one or more parameter values from the business objects;
 - associating, using the actions, the business object parameter values with a widget selected from among the one or more widgets;
 - associating the selected widget with a panel selected from the one or more panels, wherein the selected widget is arranged into a specified layout within the selected panel; and
 - generating at runtime an HTML user interface page that includes the selected panel; wherein at least one of the widgets has the capability of representing properties of the business objects as HTML.
3. (original) A method as recited in Claim 2, wherein the business object parameters are associated with one of the widgets based on the user request.
4. (original) A method as recited in Claim 2, wherein the application program is a network management application program.

5. (original) A method as recited in Claim 2, wherein receiving one or more business objects that define functions of the application program comprises receiving an XML file that defines the business objects and one or more of the parameters for the business objects.
6. (original) A method as recited in Claim 2, further comprising the step of generating, using the widget, client-side executable program code that performs one or more data validation or access control operations on user input for the user operation.
7. (currently amended) A method as recited in Claim 2, ~~wherein the step of receiving a user request comprises receiving a user request from the browser and dispatching the user request to one or the actions~~, wherein the actions interact with the business objects through service object module interfaces that provide parameter values for the business objects to the actions.
8. (original) A method as recited in Claim 2, further comprising the steps of:
receiving user input in a field of the user interface that is associated with the widget,
wherein the user input is received in HTML elements of an HTTP request from a browser;
converting the user input from the HTML elements into one or more programmatic objects having an appropriate data type for use by the application program.
9. (original) A method as recited in Claim 2, further comprising the step of associating a first widget with a second widget, wherein the first widget and second widget are related by a containment hierarchy.
10. (original) A method as recited in Claim 2, wherein each of the widgets represents one or more properties of the business objects by an HTML element.

11. (original) A method as recited in Claim 2, wherein the step of generating an HTML user interface page that includes the panel further comprises generating an HTML user interface page that includes one or more of JSP files, static HTML elements, style sheets, or images.

12. (currently amended) A method of automatically generating a consistent user interface for a network management application program, the method comprising the computer-implemented steps of:

receiving one or more business objects that each define a user action for the application program;

receiving one or more metadata elements defining parameters for the user actions of the business object;

invoking a controller that is communicatively coupled to one or more actions, widgets, and panels;

receiving a user request from a user through a browser used to interact with the application program and dispatching the user request to one or more of the actions;

the controller determining which of the one or more actions is responsible for acting on the user request;

obtaining, using the actions, one or more parameter values from the business objects;

associating, using the actions, the business object parameter values with a widget selected from among the one or more widgets;

associating the selected widget with a panel selected from the one or more panels,

wherein the selected widget is arranged into a specified layout within the selected panel; and

generating at runtime an HTML user interface page that includes the selected panel;

wherein at least one of the widgets has the capability of representing properties of the business objects as HTML.

13. (currently amended) A computer-readable storage medium carrying one or more sequences of instructions for generating a consistent user interface for an application program, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

receiving one or more business objects that each define a user action for the application program;

receiving one or more metadata elements defining parameters for the user actions of the business object;

invoking a controller that is communicatively coupled to one or more actions, widgets, and panels;

receiving a user request from a user through a browser used to interact with the application program and dispatching the user request to one or more of the actions;

the controller determining which of the one or more actions is responsible for acting on the user request;

obtaining, using the actions, one or more parameter values from the business objects;

associating, using the actions, the business object parameter values with a widget selected from among the one or more widgets;

associating the selected widget with a panel selected from the one or more panels,

wherein the selected widget is arranged into a specified layout within the selected panel; and

generating at runtime an HTML user interface page that includes the selected panel;

wherein at least one of the widgets has the capability of representing properties of the business objects as HTML.

14. (previously presented) A computer-readable storage medium as recited in Claim 13, wherein the business object parameters are associated with one of the widgets based on the user request.

15. (previously presented) A computer-readable storage medium as recited in Claim 13, wherein the application program is a network management application program.

16. (previously presented) A computer-readable storage medium as recited in Claim 13, wherein receiving one or more business objects that define functions of the application program comprises receiving an XML file that defines the business objects and one or more of the parameters for the business objects.

17. (previously presented) A computer-readable storage medium as recited in Claim 13, further comprising instructions for performing the step of generating, using the widget, client-side executable program code that performs one or more data validation or access control operations on user input for the user operation.

18. (currently amended) A computer-readable storage medium as recited in Claim 13, ~~wherein the step of receiving a user request comprises receiving a user request from the browser and dispatching the user request to one or the actions,~~ wherein the actions interact with the business objects through service object module interfaces that provide parameter values for the business objects to the actions.

19. (previously presented) A computer-readable storage medium as recited in Claim 13, further comprising instructions for performing the steps of:

receiving user input in a field of the user interface that is associated with the widget,

wherein the user input is received in HTML elements of an HTTP request from a browser;

converting the user input from the HTML elements into one or more programmatic objects having an appropriate data type for use by the application program.

20. (previously presented) A computer-readable storage medium as recited in Claim 13, further comprising instructions for performing the step of associating a first widget with a second widget, wherein the first widget and second widget are related by a containment hierarchy.

21. (previously presented) A computer-readable storage medium as recited in Claim 13, wherein each of the widgets represents one or more properties of the business objects by an HTML element.

22. (previously presented) A computer-readable storage medium as recited in Claim 13, wherein the step of generating an HTML user interface page that includes the panel further comprises generating an HTML user interface page that includes one or more of JSP files, static HTML elements, style sheets, or images.

23. (currently amended) An apparatus for generating a consistent user interface for an application program, comprising:

means for receiving one or more business objects that each define a user action for the application program;

means for receiving one or more metadata elements defining parameters for the user actions of the business object;

means for invoking a controller that is communicatively coupled to one or more actions, widgets, and panels;

means for receiving a user request from a user through a browser used to interact with the application program and dispatching the user request to one or more of the actions, wherein the controller determines which of the one or more actions is responsible for acting on the user request;

means for obtaining, using the actions, one or more parameter values from the business objects;

means for associating, using the actions, the business object parameter values with a widget selected from among the one or more widgets;

means for associating the selected widget with a panel selected from the one or more panels, wherein the selected widget is arranged into a specified layout within the selected panel; and

means for generating at runtime an HTML user interface page that includes the selected panel;

wherein at least one of the widgets has the capability of representing properties of the business objects as HTML.

24. (original) An apparatus as recited in Claim 23, wherein the business object parameters are associated with one of the widgets based on the user request.

25. (original) An apparatus as recited in Claim 23, wherein the application program is a network management application program.

26. (original) An apparatus as recited in Claim 23, wherein the means for receiving one or more business objects that define functions of the application program comprises means for receiving an XML file that defines the business objects and one or more of the parameters for the business objects.

27. (original) An apparatus as recited in Claim 23, further comprising means for generating, using the widget, client-side executable program code that performs one or more data validation or access control operations on user input for the user operation.

28. (currently amended) An apparatus as recited in Claim 23, ~~wherein the means for receiving a user request comprises means for receiving a user request from the browser and~~

~~dispatching the user request to one or the actions~~, wherein the actions interact with the business objects through service object module interfaces that provide parameter values for the business objects to the actions.

29. (original) An apparatus as recited in Claim 23, further comprising:
means for receiving user input in a field of the user interface that is associated with the widget, wherein the user input is received in HTML elements of an HTTP request from a browser;
means for converting the user input from the HTML elements into one or more programmatic objects having an appropriate data type for use by the application program.
30. (original) An apparatus as recited in Claim 23, further comprising means for associating a first widget with a second widget, wherein the first widget and second widget are related by a containment hierarchy.
31. (original) An apparatus as recited in Claim 23, wherein each of the widgets represents one or more properties of the business objects by an HTML element.
32. (original) An apparatus as recited in Claim 23, wherein the means for generating an HTML user interface page that includes the panel further comprises means for generating an HTML user interface page that includes one or more of JSP files, static HTML elements, style sheets, or images.
33. (currently amended) An apparatus for generating a consistent user interface for an application program, comprising:

a network interface that is coupled to the data network for receiving one or more packet flows therefrom;

a processor;

one or more stored sequences of instructions which, when executed by the processor, cause the processor to carry out the steps of:

receiving one or more business objects that each define a user action for the application program;

receiving one or more metadata elements defining parameters for the user actions of the business object;

invoking a controller that is communicatively coupled to one or more actions, widgets, and panels;

receiving a user request from a user through a browser used to interact with the application program and dispatching the user request to one or more of the actions;

the controller determining which of the one or more actions is responsible for acting on the user request;

obtaining, using the actions, one or more parameter values from the business objects;

associating, using the actions, the business object parameter values with a widget selected from among the one or more widgets;

associating the selected widget with a panel selected from the one or more panels, wherein the selected widget is arranged into a specified layout within the selected panel; and

generating at runtime an HTML user interface page that includes the selected panel;

wherein at least one of the widgets has the capability of representing properties of the business objects as HTML.

34. (original) An apparatus as recited in Claim 33, wherein the business object parameters are associated with one of the widgets based on the user request.
35. (original) An apparatus as recited in Claim 33, wherein the application program is a network management application program.
36. (original) An apparatus as recited in Claim 33, wherein receiving one or more business objects that define functions of the application program comprises receiving an XML file that defines the business objects and one or more of the parameters for the business objects.
37. (original) An apparatus as recited in Claim 33, further comprising instructions for performing the step of generating, using the widget, client-side executable program code that performs one or more data validation or access control operations on user input for the user operation.
38. (currently amended) An apparatus as recited in Claim 33, ~~wherein the instructions for performing the step of receiving a user request comprise instructions for performing the step of receiving a user request from the browser and dispatching the user request to one or the actions,~~ wherein the actions interact with the business objects through service object module interfaces that provide parameter values for the business objects to the actions.
39. (original) An apparatus as recited in Claim 33, further comprising instructions for performing the steps of:
- receiving user input in a field of the user interface that is associated with the widget,
 - wherein the user input is received in HTML elements of an HTTP request from a browser;
 - converting the user input from the HTML elements into one or more programmatic objects having an appropriate data type for use by the application program.

40. (original) An apparatus as recited in Claim 33, further comprising instructions for performing the step of associating a first widget with a second widget, wherein the first widget and second widget are related by a containment hierarchy.
41. (original) An apparatus as recited in Claim 33, wherein each of the widgets represents one or more properties of the business objects by an HTML element.
42. (original) An apparatus as recited in Claim 33, wherein the instructions for performing the step of generating an HTML user interface page that includes the panel further comprises generating an HTML user interface page that includes one or more of JSP files, static HTML elements, style sheets, or images.
43. (previously presented) The system of Claim 1, wherein one or more of the widgets are capable of automatically generate executable code.
44. (previously presented) The system of Claim 1, wherein one or more of the widgets are capable of performing data validation.
45. (previously presented) The system of Claim 1, wherein one or more of the widgets are arranged into a panel class.